

# Sentinel Maps

DEFENDING OUR DIGITAL WAY OF LIFE

# Recap

An Array is a list in JavaScript

```
let partyPeople = ["Spongebob", "Patrick", "Sandy"]
```

To iterate through the array

```
for (const person of partyPeople) {  
  sendInvite(person)  
}
```

# Recap

Arrays are zero indexed

```
let partyPeople = ["Spongebob", "Patrick", "Sandy"]  
partyPeople[0] // Notice, Arrays are zero-indexed!
```



Values can be changed

```
partyPeople[2] = "Squidward"
```

# Recap

**pop** – remove and return the last item

```
partyPeople.pop()
```

**shift** – remove and return the first item

```
partyPeople.shift()
```

**unshift** – insert an item to the beginning

```
partyPeople.unshift("Gary")
```





# Maps



# Learning Objective

Understand what a map is

Learn about **key** – **value** pairs

Make your very own cipher encryption

# Goal

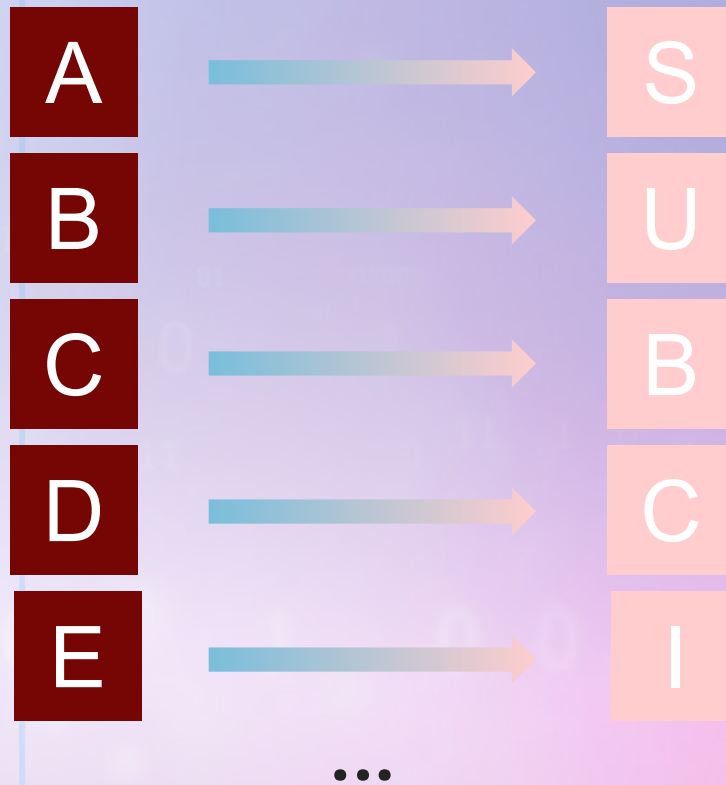
At the end of this lesson we want to be able to implement our very own mixed alphabet cipher encryptor/decryptor!



I'm a pro!

# Reminder: Mixed Alphabet Cipher


Create a **mapping** between each letter and a different letter






# Example

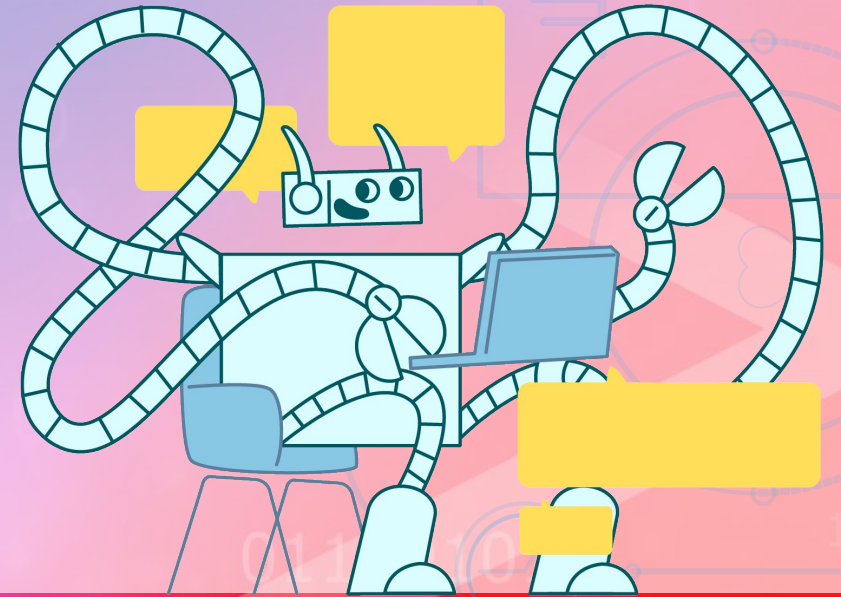
“ATTACK AT DAWN”



<b>A</b>	<b>T</b>	<b>C</b>	<b>K</b>	<b>D</b>	<b>W</b>	<b>N</b>
E	L	M	I	S	N	D



“ELLEMI EL SEND”



# Coding the Cipher

We would need to:

1. Define a mapping from each letter to another

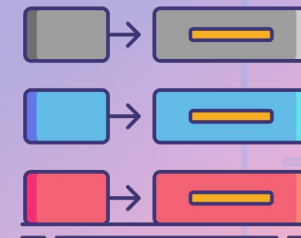
2. Perform the substitution for the letters



# Maps in JavaScript

We can initialize a map using **Key-Value pairs**

```
let myMap = { key: "value" }
```



Then to get the value of a key:

```
myMap.key  
myMap[ "key" ]
```



# Maps in JavaScript

The keys must be ✨unique ✨!

There cannot be two keys with the same name.



The keys can be anything, not just strings.

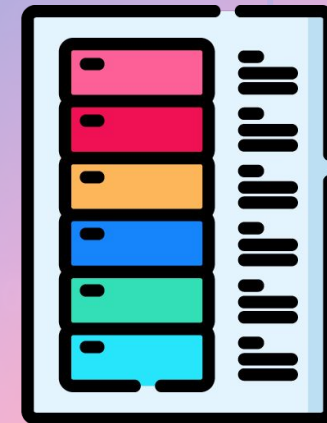


# Why Maps?

Maps allow us to **organize information** in an easy to access way

Like the index of a book:

```
{ introduction: 10, chapter1: 15, ... }
```

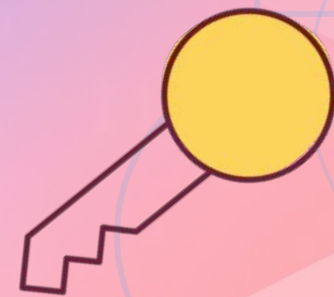


# Adding/Modifying Values

You can attach a new value for a key using the = operator

```
myMap.key = "value"  
myMap["firstName"] = "Toby"  
myMap.firstName  
// "Toby"
```

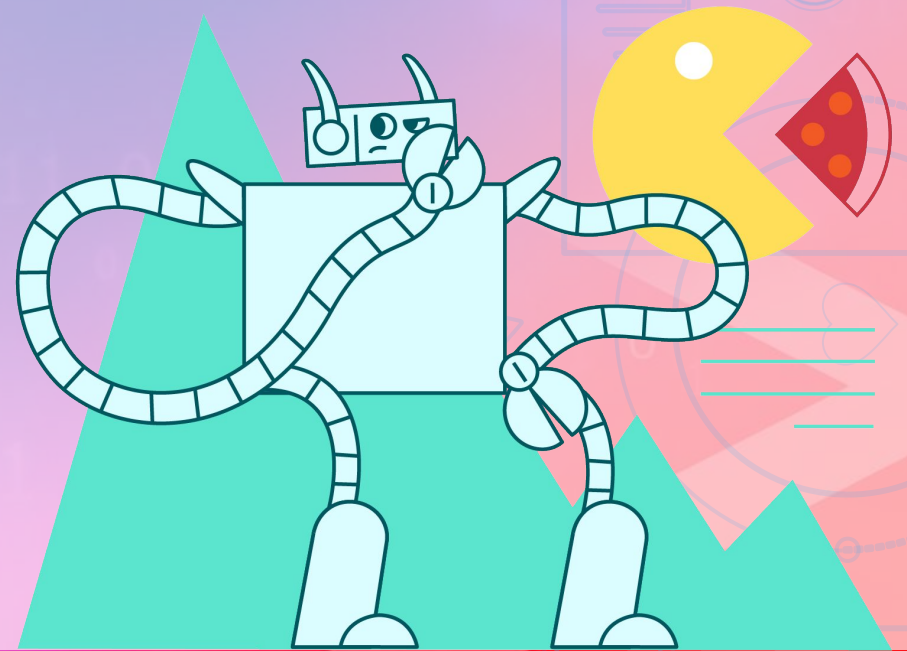
Toby



# Checking if a key exists

To check if a key exists use the following syntax:

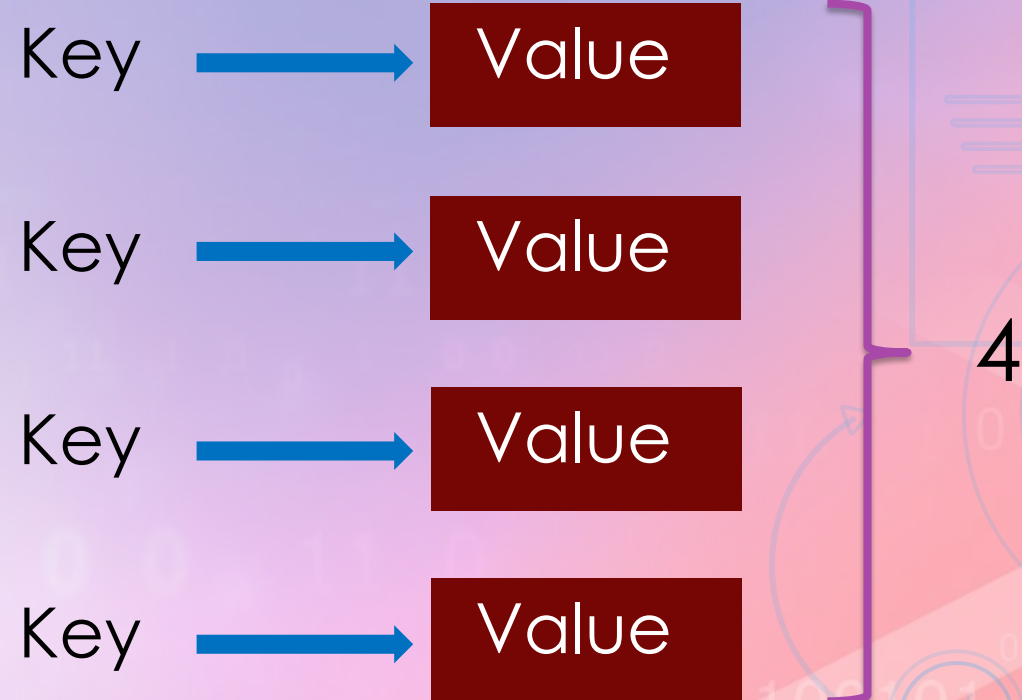
```
if (key in map) {  
    //logic  
}
```



# Map Size (Extra Reading)

To get the number of Key-Value pairs in a Map:

```
Object.keys(map).length
```





# Iterating over Map Keys

```
let birthdays = {  
  John: "09/10/1940",  
  Paul: "18/06/1942",  
  Ringo: "07/07/1940",  
  George: "25/02/1943"  
}
```

```
for (let person in birthdays) {  
  console.log(`${person} was born on ${birthdays[person]}`)  
}
```



# Iterating over Map Keys

```
let birthdays = {  
  John: "09/10/1940",  
  Paul: "18/06/1942",  
  Ringo: "07/07/1940",  
  George: "25/02/1943"  
}
```

```
for (let person in birthdays) {  
  console.log(`${person} was born on ${birthdays[person]}`)  
}
```



# Iterating over Map Pairs

Object.entries(***map***) returns an array of [key, value] pairs

The pairs themselves are just Arrays with 2 elements

```
for (const [person, date] of Object.entries(birthdays)) {  
  console.log(`${person} was born on ${date}`)  
}
```

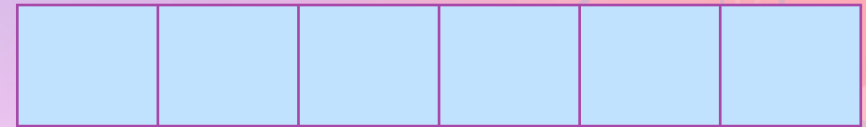
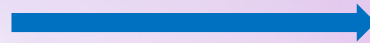
# Caveats

Maps don't have a particular **order** to their keys

So, to sort the pairs we would need to convert the Map into an Array first



Map



Array

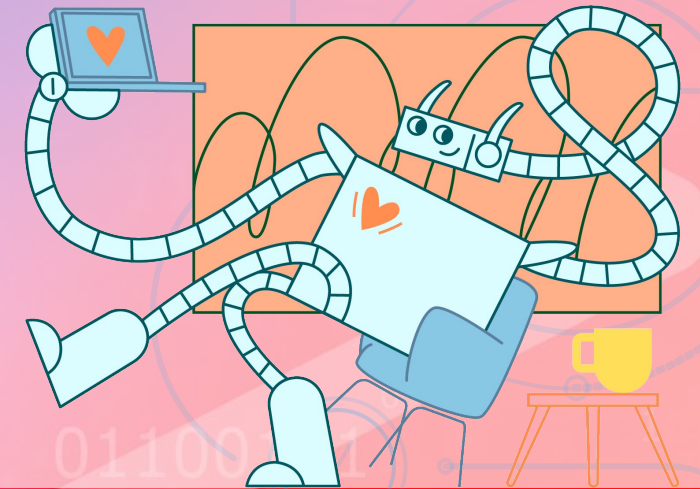


# Back to the Cipher

How can we define the mixed alphabet map?

```
const KEY = { A: "e", T: "l", C: "m", ... }
```

A	T	C	K	D	W	N
E	L	M	I	S	N	D



# Substituting

```
let text = "attack at dawn"  
let ciphertext = text.toUpperCase()  
for (const [key, value] of Object.entries(KEY)) {  
  ciphertext = ciphertext.replaceAll(key, value)  
}
```



# Summary

How do we code a cipher?

1. Define a mapping from each letter to another
2. Perform the substitution for the letters



# Summary

Initialise Map	<pre>let myMap = { key: "value" }</pre>
Check if key exist	<pre>if (key in map) {   // logic }</pre>
Replace the value	<pre>myMap.key = "value" myMap["firstName"] = "Toby"</pre>



# Summary

## Iterating over Map Keys

```
for (let person in birthdays) {  
  console.log(`${person} was born on ${birthdays[person]}`)  
}
```

## Iterating over Map Pairs

```
for (const [person, date] of Object.entries(birthdays)) {  
  console.log(`${person} was born on ${date}`)  
}
```

# Demo Time

Mixed Alphabet Cipher  
Implementation

TEXT

ICONS

HEADER

SLIDER



# Questions?

# Your Turn!

> Play around, have fun, ask questions!